Towards Faithful and Meaningful Interpretable Representations

Kacper Sokol and **Peter Flach**
Department of Computer Science, University of Bristol, Bristol, UK
{K.Sokol, Peter.Flach}@bristol.ac.uk

Abstract

Interpretable representations are the backbone of many black-box explainers. They translate the low-level data representation necessary for good predictive performance into high-level human-intelligible concepts used to convey the explanation. Notably, the explanation type and its cognitive complexity are directly controlled by the interpretable representation, allowing to target a particular audience and use case. However, many explainers that rely on interpretable representations overlook their merit and fall back on default solutions, which may introduce implicit assumptions, thereby degrading the explanatory power of such techniques. To address this problem, we study properties of interpretable representations that encode presence and absence of human-comprehensible concepts. We show how they are operationalised for tabular, image and text data, discussing their strengths and weaknesses. Finally, we analyse their explanatory properties in the context of tabular data, where a linear model is used to quantify the importance of interpretable concepts.

1 Introduction

Interpretable Representations (IRs) are the foundation of many explainability methods for black-box Machine Learning (ML) [Friedman *et al.*, 2008; Ribeiro *et al.*, 2016; Lundberg and Lee, 2017]. They facilitate translating the "language" of ML models – such as raw feature values and their complex embeddings – into concepts that are intelligible and relatable for humans. IRs create an *interface* between a computer-readable encoding of a phenomenon (collected data) and cognitively digestible chunks of information. By personalising an IR we can adjust the complexity of the resulting explanations and tune them towards a particular *audience*, making the underlying explainer more versatile and appealing.

An IR of images, for example, can be created with super-pixel segmentation, i.e., partitioning images into non-overlapping *segments*, each one representing an object of interest or pieces thereof. Similarly, text can be split into *tokens* denoting individual words, their stems or collections of words, which are not necessarily adjacent. Tabular data containing numerical features can be discretised to capture meaningful patterns, e.g., people in different age groups. In particular, these representation changes facilitate explainability of sensory data and allow retrofitting explainers into pre-existing black-box ML models [Ribeiro *et al.*, 2016].

An interpretable representation, however, is just one of many components needed to build explainability algorithms. Given high complexity of such end-to-end explainers, many of them opt for generic (thus versatile) IRs, focusing on the overall performance of the explainers and not delving into selection and optimisation of their individual components. Understandably, these explainers seek to automate the whole process, which requires IRs that can be generated without human guidance, thereby enabling their creation and evaluation at scale. Such examples are quantile discretisation for numerical features of tabular data, edge-based super-pixel segmentation for images (e.g., quick shift [Vedaldi and Soatto, 2008]), and whitespace-based tokenisation for text. This design choice can be easily justified since creating an IR that is intelligible and meaningful is often user- and application-dependant or even unique to the explained data point, therefore scaling it up is impractical without a concrete deployment use case. However, the core premise of interpretable representations is encoding concepts that are *meaningful* to the target audience, thus relying upon computer-generated IRs without understanding their behaviour and properties may be counterproductive.

Interpretable representations are often paired with a simple and inherently transparent model to form a surrogate explainer such as Local Interpretable Model-agnostic Explanations (LIME [Ribeiro *et al.*, 2016]). These explainers are very attractive since they are post-hoc, model-agnostic and – due to the use of IRs – compatible with any data type. Their versatility, however, comes at a cost: they are complex entities suffering from overparameterisation, which often manifests itself in multiple contributing sources of randomness and low fidelity of the resulting explanations [Zhang *et al.*, 2019; Rudin, 2019; Lakkaraju *et al.*, 2019; Lakkaraju and Bastani, 2020]. Some of these deficiencies can be attributed to a misuse of the interpretable representation, which can make or break an explainer [Sokol *et al.*, 2019a]. These problems can be magnified, or even render the explainer unusable, by certain pairings of interpretable representations and surrogate

model types, especially when the implicit assumptions behind both of these components are at odds. This area of research is vastly under-explored for IRs on their own and as a part of an explainer, potentially leading to sub-optimal design choices and inadequate explanations.

Explainee-driven interactive personalisation of IRs is an interesting avenue of research in this direction on the crossroads of interpretable ML and Human–Computer Interaction [Sokol and Flach, 2020c]. It has the potential to formulate guidelines for the design and operationalisation of IRs for individual applications, but such a solution comes at the expense of a user-in-the-loop architecture that cannot be automated and scaled. A more impactful research direction, which we focus on in this paper, is *automatic* creation of IRs that encode (computationally) meaningful concepts. Notably, an interpretable representation defines the question that the explanation answers and restricts the types of explanation that can meaningfully communicate this information, e.g., importance of interpretable concepts, counterfactuals or what-if statements. By understanding characteristics and behaviour of each interpretable representation and its influence on the resulting explanation – both on its own and in conjunction with a particular type of a surrogate model – we can uncover the theoretical properties of such explainers and assess their applicability and usefulness for a problem at hand.

However, the important task of choosing an appropriate interpretable representation is often overlooked in the literature. It is common to assume that an IR is given or to reuse one that was introduced in prior work without much afterthought or deliberation on its suitability, (often implicit) assumptions, properties and caveats [Laugel *et al.*, 2018; Zhang *et al.*, 2019; Lakkaraju and Bastani, 2020]. These approaches result in a uniform landscape of IRs, where the dominating objective is to understand *how important* a particular interpretable component is for a black-box prediction. Nonetheless, this type of explainability-driven sensitivity analysis comes with many unaddressed issues. For example, to discern how a particular interpretable component influences a black-box prediction, it needs to be "removed" and the resulting change in the model's prediction quantified. Most black-box models, however, cannot predict incomplete instances, especially for tabular and image data, in which case this procedure becomes ill-defined and replaced with a proxy – such as segment occlusion for images – leading to biased and untrustworthy explanations.

In this paper, we investigate capabilities and limitations of the most common type of interpretable representations – where presence and absence of interpretable concepts is encoded with a binary vector – for images and text in Section 2, and tabular data in Section 3. Our findings are a stepping stone towards building custom IRs that can be generated automatically while still representing (computationally) meaningful concepts. Among others, we discuss an implicit assumption of the *explanation locality* that is detrimental to its completeness, and a *non-bijective transformation* between the original and interpretable domains that introduces unnecessary randomness, thereby reducing fidelity and soundness of resulting explanations [Sokol *et al.*, 2019a; Sokol and Flach, 2020a]. Moreover, we investigate implicit assumptions and consequences of using proxies when it is impossible to remove information from data as imposed by the IR. We support our claims with a range of experimental results – presented in Appendix A – which illustrate how such proxies may results in *volatile explanations* that are easily influenced by the segmentation granularity and occlusion colour for images.

Furthermore, in Section 4 we examine how the choice of a surrogate model influences the resulting explanations in view of the properties and limitations of the underlying IR. To this end, we analyse tabular data with numerical features in a surrogate explainer setting, where importance-based explanations are determined by the coefficients of a linear model. In particular, we illustrate limited explanatory capabilities of an interpretable representation built upon discretisation of continuous attributes when paired with Ordinary Least Squares (OLS), a detailed derivation of which is presented in Appendix C. Such explainers lose precise encoding of the black-box decision boundary and can be manipulated by altering the distribution of the data sample used to train the OLS, which undermine reliability of the resulting explanations. As a solution, we propose using decision trees to both partition (discretise) the data space and generate explanations, e.g., with counterfactuals [van der Waa *et al.*, 2018] – a recommendation that we support with an array of experimental results presented in Appendix B. Finally, we conclude our work and discuss future research directions in Section 5.

The choice of the IR and surrogate model for our analysis is motivated by the popularity of these particular approaches in the literature. Namely, LIME [Ribeiro *et al.*, 2016] and RuleFit [Friedman *et al.*, 2008] use a surrogate linear model to estimate importance of interpretable concepts. Additionally, LIME and SHapley Additive exPlanations (SHAP [Lundberg and Lee, 2017]) employ an interpretable representation that encodes presence and absence of intelligible concepts to formulate their explanations. Friedman *et al.* (2008), on the other hand, experimented with automatically learning a more complex IR by training a random forest and extracting rules from therein. These logical rules, which express various concepts, are then used as binary features for training a linear model, thus improving the expressiveness of the interpretable representation. More recently, Garreau and Luxburg (2020) analysed theoretical properties and parameterisation of vanilla LIME for tabular data, including its interpretable representation and surrogate linear model, however their work treated the explainer as an end-to-end algorithm and operated under quite restrictive assumptions, e.g., linearity of the black-box model.

2 Sensory Interpretable Representations

The operationalisation of interpretable representations vary for different data types – tabular, image and text – but their machine representation is usually consistent: a binary vector indicating presence (1) or absence (0) of certain human-understandable concepts for a selected data point. To this end, tabular data with numerical features are discretised, creating a hyper-rectangle partition of the space. The binary interpretable representation is unique to each hyper-rectangle,

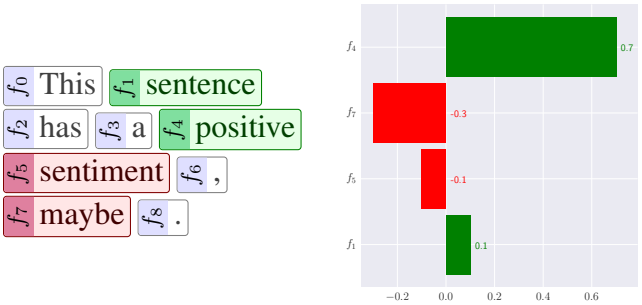
hence one of them has to be selected as the explanation target. The binary IR of an arbitrary data point is then computed by juxtaposing the discrete encoding of the hyper-rectangle it belongs to against the explained hyper-rectangle. This operation results in a binary on/off vector that for each discrete dimension indicates whether the selected data point lies in the same partition as the explained hyper-rectangle or not – see Section 3 for more details.

The interpretable representations of image and text data, on the other hand, come more naturally and share many properties. Images are partitioned into non-overlapping segments called super-pixels, which are then represented in the interpretable binary space as either present or absent. Similarly, text is split into tokens that can encode individual words, their stems or collections of words, which presence or absence is encoded in the IR. These two interpretable representations are relatively easy to generate automatically and, when configured correctly, encode (computationally) meaningful concepts. High-dimensionality of raw data does not affect their comprehensibility, which is the case with tabular data where we are generally confined to three dimensions given the inherent spatio-temporal limitations of our visual system. Moreover, dimensionality reduction for images and text is unnecessary or even harmful: removing super-pixels from images is an ill-defined procedure and would results in "holes", whereas for text it can be used to discard stop words and punctuation, but such pruning is often incorporated into the preceding tokenisation step.

2.1 Text

The interpretable domain based on presence and absence of tokens in text is very natural and appealing to humans. Individual words and their groups encode understandable concepts and their absence may alter the meaning of a sentence, which reflects how humans comprehend text. A naïve IR can represent text as a bag-of-words, where each word becomes a token, thereby forfeiting the influence of word ordering and the information carried by their co-appearance. We can easily improve upon that and capture the dependencies between words by including n-gram groupings. Applying other pre-processing steps, e.g., extracting word stems, can also be beneficial for the human-comprehensibility of such interpretable representations. Machine processing of text is a well-established research field [Manning and Schutze, 1999], providing plenty of inspiration for designing appealing IRs.

Once text is pre-processed and tokenised, it is *deterministically* transformed into the binary interpretable representation. To this end, a sentence is encoded as a Boolean vector of length equal to the number of tokens in the IR, where 1 indicates presence of a given token and 0 its absence, meaning that the original sentence is encoded with an all-1 vector. By flipping some components of this vector to 0, we effectively remove tokens from this sentence and create its variations. Notably, high dimensionality of this representation does not undermine the comprehensibility of the resulting explanations since altered text cannot have more tokens than the original sentence. Explanations based on token importance can be overlaid on top of text by highlighting each token with a different shade of green (positive influence) or red (negative influence) given their respective importance for the explained class – see Figure 1 for an example.



(a) Words are the components of the interpretable representation.

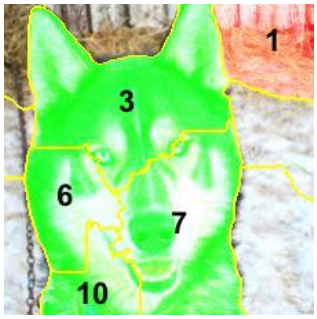
(b) Explanation shows influence of the interpretable components.

Figure 1: Example of an importance-based explanation of text with a *bag-of-words* interpretable representation. Panel (a) illustrates a sentence whose (positive) *sentiment* is being decided by a black-box model. The colouring of each word in panel (a) conveys its influence on the prediction, with panel (b) depicting their magnitudes.

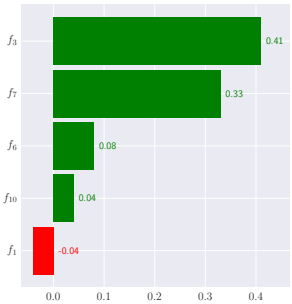
2.2 Images

The interpretable representation of image data operates similarly – see Figure 2. Images are algorithmically segmented into super-pixels, often using edge-based methods [Ribeiro *et al.*, 2016] such as quick shift [Vedaldi and Soatto, 2008], but the resulting partition may not convey (cognitively) meaningful concepts from a human perspective. *Semantic segmentation* or outsourcing this task to the user usually yields better results [Sokol and Flach, 2020b; Sokol and Flach, 2020c]. Next, the segments are represented as a binary vector indicating presence (1) or absence (0) of information in each super-pixel, where an all-1 vector corresponds to the original image. However, removing a super-pixel from an image when setting one of the interpretable components to 0 is an ill-defined procedure. The most appealing and semantically-meaningful solution would be to "remove" the content of a segment by occluding it with another object, akin to Benchmarking Attribution Methods (BAM [Yang and Kim, 2019]), or retouching it in a context-aware manner, e.g., with what is anticipated in the background, thus preserving the colour continuity of the image. Both of these approaches are intuitive, but they are difficult to automate and scale since they are mostly limited to image partitions where each super-pixel represents a self-contained and semantically-coherent object.

Instead, a computationally-feasible proxy is commonly used to hide the information carried by the super-pixels: segments are occluded with a solid colour. However, this approach comes with its own implicit assumptions and limitations, which are often ovelooked. For example, LIME uses the mean colour of each segment to mask its content [Ribeiro *et al.*, 2016] – see Figure 3a – which for some segmentation and colour distribution of an image may undermine the utility of the occlusion procedure. With this approach, segments that have a relatively uniform colour gamut may, effectively, be impossible to occlude; this is especially common for segments that are in the background or out of focus,



(a) Image segments are the components of the interpretable representation.



(b) Explanation shows influence of the IR components on *Eskimo dog* prediction.

Figure 2: Example of an importance-based explanation of image data with the interpretable representation built upon *segmentation*. Panel (a) illustrates an image that is being classified by a black-box model. The colouring of each super-pixel in panel (a) conveys its influence on the prediction, with panel (b) depicting their magnitudes.

(a) Mean-colour occlusion.

(b) Black occlusion.

Figure 3: Image occlusion strategy influences the resulting explanations (see Appendix A). The picture shown in Figure 2a is classified by a black box as *Eskimo dog* with 83% probability. Mean-colour occlusion of all the segments but one (a) results in 77% and black occlusion (b) in 9% probability of the same class, showing that the former approach cannot effectively remove information from this particular image.

e.g., bokeh and depth-of-field effects. Furthermore, whenever the segmentation coincides with objects' edges or regions of an image where colour continuity is not preserved, which is common for edge-based segmenters, occluding super-pixels with their mean colour causes (slight) colour variations between adjacent segments. These artefacts preserve edges in a (partially) occluded image, and they often convey enough information for a black-box model to correctly recognise its class, for example see Figure 3. Segmentation granularity is also important: the smaller the segments are, the more likely it is that their colour composition is uniform given the "continuity" of images, i.e., high correlation of adjacent pixels, resulting in a similar effect as above.

Since most of these issues are consequences of using the mean-colour occlusion, it may seem that fixing a single masking colour for all of the segments would eradicate some of these problems. Such an approach hides the edges between occluded super-pixels and removes their content instead of just "blurring" the image, however the edges between occluded and preserved segments remain visible. Moreover, the choice of the masking colour may impact the explanations themselves regardless of the colouring strategy. This particular type of a proxy for removing information from image segments implicitly assumes that the black-box model is *neutral* with respect to the occlusion colour, i.e., none of the modelled classes is biased towards it. Adjusting the granularity of the segmentation also plays an important role given high correlation of adjacent super-pixels. We support these observations with a range of experiments done for occlusion-based interpretable representations of images, the results of which are presented in Appendix A. In particular, they exemplify the degree to which the segmentation granularity as well as the occlusion strategy and colour affect the resulting explanations.

2.3 Discussion

The interpretable representations of image and text data discussed above are *implicitly local* – they are only valid for the data point (sentence or image) for which they were created. Another, often overlooked, property that they share is *bijectiveness* of the transformation (within the scope of a single instance): there is one-to-one correspondence between a data point and its interpretable representation and a uniquely defined inverse transformation from the interpretable to the original data domain, assuming that the image occlusion strategy (colour) and text pre-processing are fixed [Sokol *et al.*, 2019a]. This property helps to guarantee uniqueness of explanations, which is very important for their stability, hence preserving explainees' trust [Rudin, 2019; Sokol and Flach, 2020a]. Transforming between the two domains seamlessly only requires memorising the structure or skeleton of the explained data point: adjacency of segments and their original pixel values for images, and order of tokens and their pre-processing for text. Out of the two IRs, the one for text has the advantage of allowing the tokens to be *truly* removed from a sentence, however this is more of a property of the underlying predictive model rather than the IR itself – text classifiers are more flexible and do not assume input of a fixed length, while vision models cannot handle "missing" pixels.

3 Tabular Interpretable Representations

In contrast to raw pixel values and word embeddings, tabular data do not require an interpretable representation to be explainable since their features are often human-comprehensible. However, if the explanation is to answer a specific question – as was the case for images and text – using an IR may be helpful. Continuing with the sensitivity analysis setting, for tabular data we are interested in how presence and absence of certain *binary* concepts, which the explained data point exhibits, influence its prediction. One approach is to treat the specific feature values of the explained instance as concepts: if a feature value of a data point is identical to the value of the same feature for the explained instance, the concept is *present* (1), otherwise it is *absent* (0). While appealing for categorical features, considering each and every unique

value of a numerical attribute is counter-intuitive given their inherent continuity. Moreover, doing so may not reflect the underlying human thought process, e.g., "high sugar content" in contrast to "70g of sugar per 100g of a product", with both 0g and 100g in the latter case encoded as an *absent* concept in the corresponding IR. Building up on this observation, a natural extension of such an interpretable representation is to consider discretisation of numerical features as interpretable concepts: if a feature value of a data point is within the same range as the value of the same attribute for the explained instance, the concept is *present* (1), otherwise it is *absent* (0).

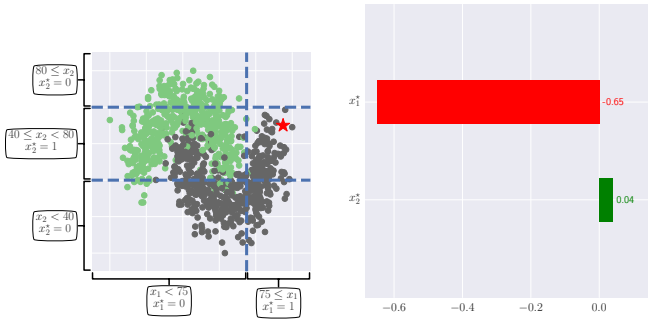
3.1 Interpretable Representation via Discretisation

Interpretable representations of tabular data are generated by preserving categorical features and discretising numerical attributes into "categorical" bins, e.g., $x_2 < 5$, $5 \leq x_2 < 7$ and $7 \leq x_2$. Next, the binary on/off representation is computed based on the data point selected to be explained, which facilitates expressing importance of each interpretable concept in relation to the black-box prediction of this instance. Since a binary representation allows to encode only 2 events for each attribute, it commonly indicates a feature value belonging to the same (1) or different (0) numerical bin as the data point selected to be explained. For example, if the second feature x_2 of the explained data point $\mathring{x}$ is $\mathring{x}_2 = 6.5$, based on the aforementioned bin boundaries, any instance x whose second attribute is within the $5 \leq x_2 < 7$ range is assigned $\hat{x}_2 = 1$ in the binary IR, and 0 otherwise. When paired with a linear surrogate model, this can be understood as investigating the importance of each feature value being within the specified range (the concept "switched on") for the black-box prediction of the selected data point, or more precisely *any* instance situated within the same hyper-rectangle – see Figure 4 for an example.

Notably, this binary interpretable representation of tabular data is specific to the explained data point (more generally, its hyper-rectangle), as was the case for images and text. Nonetheless, for tabular data the underlying discretisation can be reused for explaining any instance from the same data set. While a common practice [Ribeiro *et al.*, 2016], such an approach undermines faithfulness of the resulting explanations – the goal is to produce a *local* explanation of the selected data point, hence the discretisation should be truthful within the explained neighbourhood. Neither globally nor locally faithful discretisation can capture uniqueness of a black-box decision boundary universally well for an arbitrary data subspace [Sokol *et al.*, 2019a]. Therefore, reusing the same discretisation to generate individual IRs for tabular data can be compared to creating a super-pixel partition of a specific image and reapplying it to other, unrelated images, resulting in a technically valid, yet conceptually meaningless interpretable representation. Finally, selecting the bin boundaries when discretising tabular data is non-trivial and biases the explanation akin to the influence of the segmentation granularity and occlusion colour on image explanations.

3.2 Faithfulness

The two factors that determine quality and faithfulness of a binary interpretable representation of tabular data are the *data*



(a) Discretised and binarised numerical features become the components of the interpretable representation ($x^\star$).

(b) Explanation shows influence of IR components on predicting the *grey* class for the $\star$ instance and, more generally, the $x^\star = (1, 1)$ hyper-rectangle.

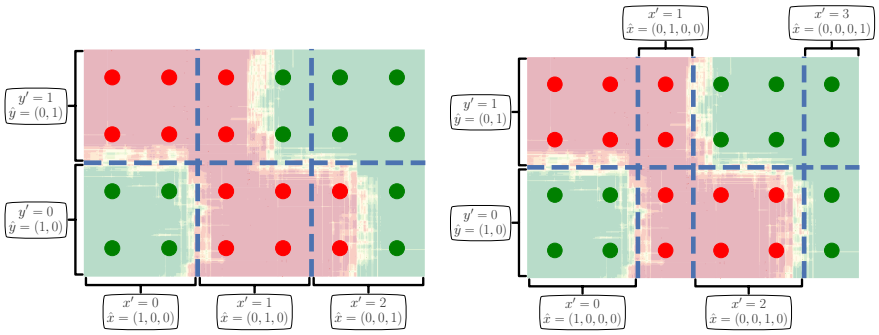
Figure 4: Example of an importance-based explanation of tabular data with the interpretable representation built upon *discretisation* and *binarisation*. Panel (a) illustrates an instance (red $\star$) that is being predicted by a black-box model. The dashed blue lines mark feature binning, grey and green dots denote two classes, and $x^\star$ encodes the binary IR created for the $\star$ data point. Panel (b) depicts the magnitude of the influence that $x_1^\star : 75 \leq x_1$ and $x_2^\star : 40 < x_2 \leq 80$ have on predicting the *grey* class for the $\star$ instance (as well as any other data point located within the same hyper-rectangle).

point selected to be explained – which specifies the reference hyper-rectangle – and the ability of the *discretisation* algorithm to locally approximate the black-box decision boundary. However, only the latter is controlled algorithmically and can either be explicitly **global**, i.e., learnt with respect to a whole data set, or **local**, thus focusing on a specific neighbourhood. Furthermore, each one can either observe just the *data distribution*, or additionally take into account their *black-box predictions*, presenting us with two distinct approaches:

distribution-aware discretisation – Figure 5a – is based on the density of data in the local or global region chosen to be explained, e.g., quantile discretisaton; and

class-aware discretisation – Figure 5b – partitions data according to a black-box decision boundary confined within the local or global region chosen to be explained.

Since the predominant role of local surrogate explanations is to approximate and simplify the behaviour of a black box near a selected instance, the latter type should be preferred. It is a stepping stone towards representing human-intelligible concepts that are coherent with predictions of the underlying model, thus producing faithful and appealing insights. However, to the best of our knowledge, class-aware discretisation approaches are absent in the explainable AI literature. Computationally, their objective can be expressed as maximising the *purity* or *uniformity* of each hyper-rectangle with respect to the black-box predictions of data that it encloses – this applies to both probabilistic and crisp models. This observation suggests that **learning interpretable representations with decision trees** – specifically, the data space partitions that they create – is an algorithmically sound approach to optimising the aforementioned objective, which we shown exper-



(a) *Distribution*-aware discretisation.

(b) *Class*-aware (predictions) discretisation.

Figure 5: Discretisation is the main building block of interpretable representations of tabular data. It can either be learnt based on data features alone – panel (a) – or additionally consider their black-box predictions (background shading) – panel (b).

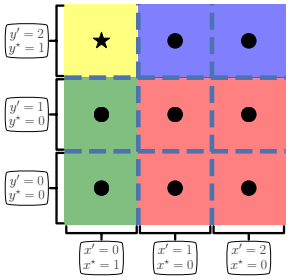


Figure 6: Some hyper-rectangles – created with discretisation – become indistinguishable in the binary interpretable representation of tabular data. The $\star$ indicates the explained instance and the background shading marks unique binary encodings.

imentally in Appendix B. While tree training procedures tend to be greedy, alternatives that consider multiple features at any given iteration could improve the quality of the resulting IRs even further.

3.3 Information Loss and Removal

The intermediate discrete representation of a tabular IR uniquely encodes each hyper-rectangle – see the (x', y') coordinates in Figure 6. However, the same is not true for the *binary* interpretable representation if any categorical feature has more than two unique values or any numerical attribute is partitioned into more than two intervals. In such cases, the transformation between these two representations loses information as depicted by the background shading and the $(x^\star, y^\star)$ coordinates in Figure 6. For each of these binary features, 1 is assigned to the partition that contains the explained data point and 0 to all the other intervals, effectively making them indistinguishable. Similarly, information is lost when transitioning from the original data representation into their discretised form: each hyper-rectangle, which has unique coordinates in this domain, contains multiple data points that become indistinguishable – see Figure 5 for reference. These two many-to-one transformations, which are needed to create an interpretable representation of tabular data, contribute to non-bijectiveness discussed in the following section.

The impossibility to distinguish data belonging to different hyper-rectangles in the binary interpretable representation is particularly detrimental to capturing the complexity of a black-box decision boundary. While the underlying discretisation may have closely approximated its intricacies, these can be lost when transitioning into the binary representation, especially if the decision boundary runs across hyper-rectangles that were merged in this process. For example, consider the discretisation shown in Figure 5b assuming that the explained instance resides in the $(x', y') = (1, 1)$ hyper-rectangle – top row, second from the left. In the binary representation, the remaining top-row hyper-rectangles $(0, 1)$, $(2, 1)$ and $(3, 1)$ would be grouped – akin to the process depicted by the background shading in Figure 6 – thus forfeiting the information that the first one belongs to the red class and the latter two to the green class.

Furthermore, in contrast to text and comparably to image data, *removing* information from the original representation by setting a member of the binary IR to 0 is infeasible and requires a proxy. Within our operationalisation of the interpretable representation of tabular data, switching off an interpretable concept is equivalent to placing the value of the corresponding numerical feature outside of the range encoded by this concept. Similarly, for categorical features this step is equivalent to choosing any other value not encoded by the corresponding interpretable concept. This procedure ensures diversity of data in the explained neighbourhood at a high price of introducing two sources of randomness when manipulating the IR.

3.4 Non-bijectiveness

A direct consequence of the information loss is *non-bijectiveness* of the interpretable representation transformation function, which increases volatility of the resulting explanations. Transitioning from an original into a discrete representation is a many-to-one operation if the underlying data set contains numerical features. Furthermore, transforming the discrete representation into a binary IR is also a many-to-one mapping if any discretised attribute has more than two unique values. Recall that surrogate explainers – a big beneficiary of IRs – sample data from the (binary) interpretable representation, which implicitly introduces locality of these data points and the resulting explanations. While reasonable for images and text, following the same procedure for a tabular IR entails reversing the binary sample back to the original domain, which requires *random sampling* because of the forward transformation not being bijective [Sokol *et al.*, 2019a]. To this end, we first choose at random one of the concatenated hyper-rectangles if the binary component is 0 (1 uniquely identifies a hyper-rectangle); next, we sample a numerical value from the range defined by this hyper-rectangle, e.g., using a Gaussian distribution fitted to the (training) data enclosed by this hyper-rectangle (categorical features are uniquely identified by a hyper-rectangle).

Image and text data require manipulating the binary IR

since sampling from raw text or pixels is semantically an ill-defined procedure. However, tabular data can be sampled in their original representation, which allows for an algorithmic workaround of non-bijectiveness. Data drawn from the original domain can be easily transformed into a discrete representation and followed with binarisation. But moving in the opposite direction in a deterministic fashion requires memorising the matching between these points in different representations when executing the forward transformation – this matching can be compared to storing segment adjacency for images or a sentence skeleton for text. By sampling in the original domain and binding different representations of each instance, we avoid using the ill-defined inverse IR transformation of tabular data, hence reduce randomness and improve stability of the resulting explanations. Nonetheless, this sampling strategy forfeits the implicit locality achieved by operating directly on the binary representation, therefore the substitute sampling algorithm should be *explicitly local* to capture the explained subspace in detail [Sokol *et al.*, 2019a].

4 Modelling Interpretable Representations

Interpretable representations are commonly paired with linear models [Friedman *et al.*, 2008; Ribeiro *et al.*, 2016] to explain black-box predictions with concept importance, meaning that such explanations are subject to assumptions and limitations of these models. In particular, the coefficients of linear models can be deceiving when the target variable is *non-linear* with respect to data features, the features are *co-dependant* or *correlated*, and the feature values are *not normalised* to the same range [Sokol *et al.*, 2019a; Sokol and Flach, 2020b]. Intuitively, the first two properties may not hold for high-level interpretable representations since their components are highly inter-dependent, e.g., adjacent image segments, neighbouring words and bordering hyper-rectangles, thus the resulting explanations can misrepresent the possible relations between these concepts. Friedman *et al.* (2008) addressed some of these issues by using logical rules extracted from random forests as binary interpretable concepts, which they modelled with a linear predictor, however the overlap between these rules remains the culprit of the feature independence assumption.

To overcome these limitations and facilitate explanations more diverse than importance of interpretable concepts, alternative surrogate models can be used [Sokol *et al.*, 2019a]. Logical models, such as *decision trees*, are particularly appealing given that they provide a wide range of explanations and do not introduce any restrictions on the features, albeit they impose axis-parallel partition of the feature space [Sokol and Flach, 2020b]. They are particularly suited for explaining tabular data, for which they alleviate the need for an independent interpretable representation as noted in Section 3.2. In particular, they can automatically learn a locally faithful, class-aware discretisation, with the added benefit of modelling combinations of hyper-rectangles and not suffering from information loss and non-bijectiveness [Sokol *et al.*, 2019a].

On the other hand, using linear models to capture concept importance of tabular data is particularly **flawed** when utilising the interpretable representation introduced in Section 3. The information loss suffered when transitioning from the discrete into the binary representation partially forfeits the preceding effort to faithfully capture the black-box decision boundary during the discretisation step. Appendix C demonstrates this phenomenon with an analytical solution to OLS. Notably, it shows an unexpected influence of the number of data points sampled in each hyper-rectangle on the resulting explanations (concept importance magnitudes) and irrelevance of feature partitions other than the ones directly enclosing the explained instance.

5 Conclusions and Future Work

Our findings show the importance of building semantically and computationally meaningful interpretable representations, and their role in defining the question answered by the resulting explanations. Among others, we demonstrated that generic algorithms for building IRs may be insufficient, and that the intended application domain and audience, as well as interactive customisation and personalisation, should be considered. In particular, we discussed a popular operationalisation of interpretable representations for image, text and tabular data, where they are used as binary indicators of presence and absence of interpretable concepts. This framework is then used in conjunction with surrogate models to quantify importance of such concepts for individual black-box predictions. In this setting, we identified challenges such as information removal proxies, parametrisation, faithfulness and bijectiveness – which are particularly prominent for tabular data – and showed how to overcome them. We also demonstrated the limitations of explaining binary interpretable representations of tabular data with linear models and suggested logical models as a viable alternative. Our future work will investigate algorithmic information removal proxies – focusing on meaningful occlusion techniques for images – and modelling interpretable representations with technologically-compatible approaches such as logical predictive models.

References

[Achanta *et al.*, 2012] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

[Friedman *et al.*, 2008] Jerome H Friedman, Bogdan E Popescu, et al. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.

[Garreau and Luxburg, 2020] Damien Garreau and Ulrike Luxburg. Explaining the explainer: A first theoretical analysis of LIME. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International*

Conference on Artificial Intelligence and Statistics, volume 108 of *Proceedings of Machine Learning Research*, pages 1287–1296, Online, 26–28 Aug 2020. PMLR.

[Lakkaraju and Bastani, 2020] Himabindu Lakkaraju and Osbert Bastani. "How do I fool you?" Manipulating user trust via misleading black box explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 79–85, 2020.

[Lakkaraju *et al.*, 2019] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, 2019.

[Laugel *et al.*, 2018] Thibault Laugel, Xavier Renard, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Defining locality for surrogates in post-hoc interpretablity. *3rd Workshop on Human Interpretability in Machine Learning (WHI 2018) at the 35th International Conference on Machine Learning (ICML 2018), Stockholm, Sweden*, 2018. arXiv preprint arXiv:1806.07498.

[Lundberg and Lee, 2017] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[Manning and Schutze, 1999] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8026–8037. Curran Associates, Inc., 2019.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

[Rudin, 2019] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[Sokol and Flach, 2020a] Kacper Sokol and Peter Flach. Explainability fact sheets: A framework for systematic assessment of explainable approaches. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 56–67, 2020.

[Sokol and Flach, 2020b] Kacper Sokol and Peter Flach. LIMEtree: Interactively customisable explanations based on local surrogate multi-output regression trees. *Under Review at the Artificial Intelligence Journal, Special Issue on Explainable Artificial Intelligence*, 2020. arXiv preprint arXiv:2005.01427.

[Sokol and Flach, 2020c] Kacper Sokol and Peter Flach. One explanation does not fit all. *KI-Künstliche Intelligenz*, pages 1–16, 2020.

[Sokol *et al.*, 2019a] Kacper Sokol, Alexander Hepburn, Raul Santos-Rodriguez, and Peter Flach. bLIMEy: Surrogate prediction explanations beyond LIME. *2019 Workshop on Human-Centric Machine Learning (HCML 2019) at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada*, 2019. arXiv preprint arXiv:1910.13016.

[Sokol *et al.*, 2019b] Kacper Sokol, Raul Santos-Rodriguez, and Peter Flach. FAT Forensics: A Python toolbox for algorithmic fairness, accountability and transparency. *arXiv preprint arXiv:1909.05167*, 2019.

[Sokol *et al.*, 2020] Kacper Sokol, Alexander Hepburn, Rafael Poyiadzi, Matthew Clifford, Raul Santos-Rodriguez, and Peter Flach. FAT Forensics: A Python toolbox for implementing and deploying fairness, accountability and transparency algorithms in predictive systems. *Journal of Open Source Software*, 5(49):1904, 2020.

[van der Waa *et al.*, 2018] Jasper van der Waa, Marcel Robeer, Jurriaan van Diggelen, Matthieu Brinkhuis, and Mark Neerincx. Contrastive explanations with local foil trees. *Workshop on Human Interpretability in Machine Learning (WHI 2018) at the 35th International Conference on Machine Learning (ICML 2018), Stockholm, Sweden*, 2018. arXiv preprint arXiv:1806.07470.

[van der Walt *et al.*, 2014] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: Image processing in Python. *PeerJ*, 2:e453, 6 2014.

[Vedaldi and Soatto, 2008] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *European conference on computer vision*, pages 705–718. Springer, 2008.

[Yang and Kim, 2019] Mengjiao Yang and Been Kim. Benchmark attribution methods with ground truth. *2019 Workshop on Human-Centric Machine Learning (HCML 2019) at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada*, 2019.

[Zhang *et al.*, 2019] Yujia Zhang, Kuangyan Song, Yiming Sun, Sarah Tan, and Madeleine Udell. "Why should you trust my explanation?" Understanding uncertainty in LIME explanations. *AI for Social Good Workshop at the 36th International Conference on Machine Learning (ICML 2019), Long Beach, California*, 2019. arXiv preprint arXiv:1904.12991.

A Behaviour of Occlusion-based Interpretable Representations of Images

Occlusion-based interpretable representations of images are parameterised by the segmentation *granularity* and the *colouring* strategy used for "removing" the content of super-pixels to hide them from a black box. The exact relation between these two properties and the resulting explanations is discussed in Section 2.2, with the main conclusions summarised in this appendix. We support our findings with a collection of experimental results presented in Figure 7, the technical setup of which is discussed in the following paragraph. While image IRs are influenced by both segmentation granularity and occlusion strategy (i.e., an information removal proxy), text data do not require the latter given that language models usually allow flexible input size – see Section 2.3 for more details. Studying the effect of text pre-processing and tokenisation (which correspond to image partitioning) on the quality of relevant interpretable representations, however, is a challenging task that may not provide comprehensive insights given the breadth and scope of available techniques.

Experiment Setup For all of the experiments, our black box was the pre-trained *Inception v3* neural network distributed with PyTorch [Paszke *et al.*, 2019]. We sampled 100 (square and no smaller than 256×256 pixels) test images at random from the ImageNet [Deng *et al.*, 2009] validation set, which were next resized to 256×256 pixels. We segmented these images with SLIC [Achanta *et al.*, 2012] – k-means clustering in the RGB (Red, Green Blue) colour space – using the implementation provided by scikit-image (`skimage.segmentation.slic`) [van der Walt *et al.*, 2014]. We executed our experiments using the bLIMEy algorithmic framework [Sokol *et al.*, 2019a], which modularises surrogate explainers into: *interpretable representation*, data sampling and explanation generation, and implements these building blocks within the FAT Forensics Python package [Sokol *et al.*, 2019b; Sokol *et al.*, 2020].

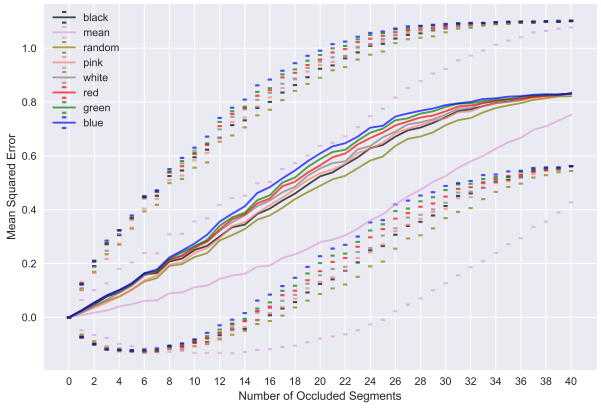
Segment occlusion was done with the following selection of colours described in the RGB space:

- black – (0, 0, 0);
- white – (255, 255, 255);
- red – (255, 0, 0);
- green – (0, 255, 0);
- blue – (0, 0, 255);
- pink – (255, 192, 203);
- mean – each super-pixel is replaced with a solid patch of the mean colour computed for the pixels residing within this segment; and
- random – a separate random colour, sampled uniformly from the RGB space, is used to occlude each individual super-pixel.

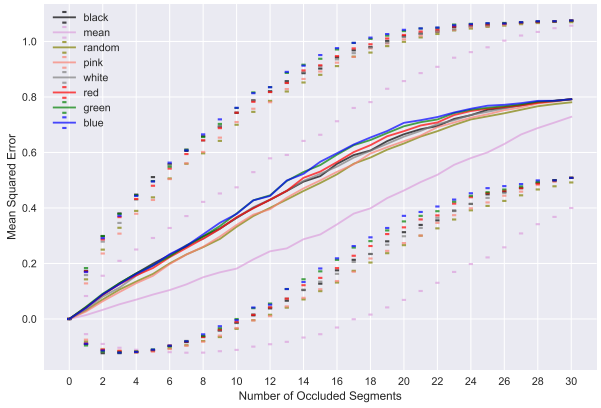
We partitioned the test images into 5, 10, 15, 20, 30 and 40 regions to capture the influence of the segmentation granularity on the IR – these tiers are visualised in separate panels of Figure 7. For a fixed number of segments, we iterated the quantity of occluded super-pixels from 0 to all of the partitions (x-axis in Figure 7), randomising the occlusion pattern 20 times at each step. We applied this procedure to all of our test images, separately for every colouring strategy. Finally, we measured the influence of each occlusion strategy and segmentation granularity by calculating the Mean Squared Error (MSE) between the probability of the top class predicted for an original image and the prediction of the same class when the image was (partially) occluded (y-axis in Figure 7).

Occlusion Colour Figure 7 provides a clear evidence that *mean*-colour occlusion strategy behaves unlike any other approach, including the *random* method. The lower MSE for this particular technique indicates that it is not as effective in "removing" class-identifying information from images as any other occlusion strategy that we tested. Intuitively, the reason for this behaviour is the "blurring" effect mentioned in Section 2.2 and exemplified in Figure 3. This phenomenon becomes more pronounced when images are segmented into smaller super-pixels (i.e., having more of them for a fixed image size) since each partition becomes more uniform with respect to the colour of its individual pixels – the increasing separation of the *mean* strategy MSE line when moving from 5 segments (Figure 7f) to 40 segments (Figure 7a).

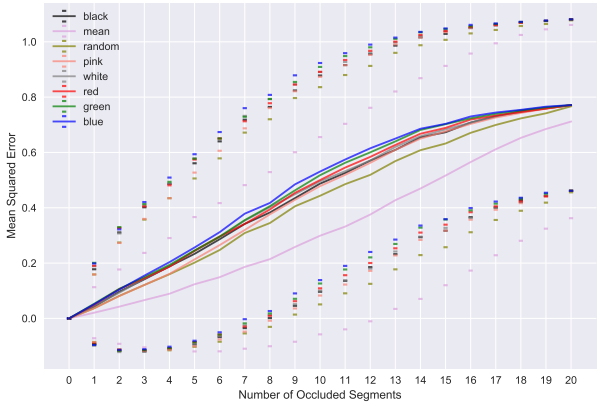
Segmentation Granularity By inspecting each panel of Figure 7, we can see that the granularity of segmentation directly affects the *mean*-colour occlusion strategy – the aforementioned separation between the MSE line of the *mean* approach and every other line. The behaviour of all the fixed-colour approaches, on the other hand, is very similar for any number of segments regardless of the exact occlusion colour – these MSE lines are clustered together in Figure 7. This observation also applies to the *random* strategy, which can be very volatile since it randomises the occlusion colour for each individual super-pixel. Both of these insights are a clear evidence that using the *mean* colouring should be avoided in occlusion-based interpretable representations of images. Figure 7 substantiates our observation that this occlusion strategy becomes less effective when the number of segments increases since relatively small segments tend to have a uniform colour distribution because of the pixel "continuity", i.e., high correlation of neighbouring pixels, making them visually similar to their respective *mean*-coloured patches. Additionally, this phenomenon may affect images that have an out-of-focus background, e.g., portraits, since their blurry regions will be difficult to "remove" with the *mean*-colour occlusion strategy – refer to Section 2.2 for an in-detail discussion.



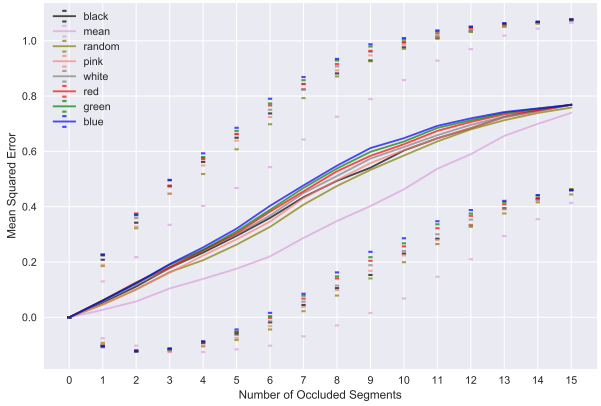
(a) Mean Squared Error for a 40-segment partition.



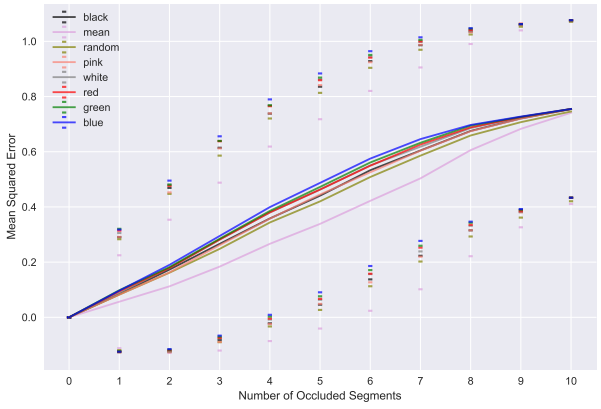
(b) Mean Squared Error for a 30-segment partition.



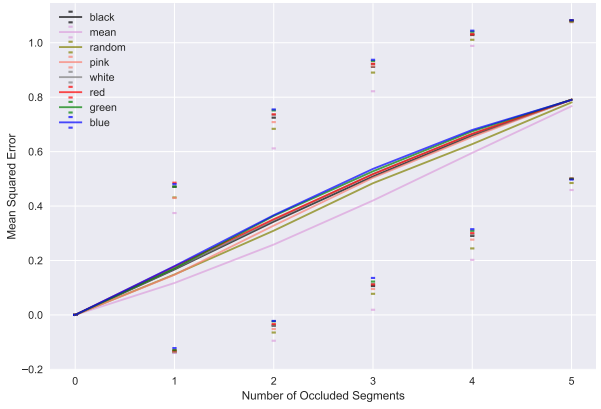
(c) Mean Squared Error for a 20-segment partition.



(d) Mean Squared Error for a 15-segment partition.



(e) Mean Squared Error for a 10-segment partition.



(f) Mean Squared Error for a 5-segment partition.

Figure 7: Mean Squared Error calculated between the top prediction of an image (probability estimate) and predictions of the same class when progressively occluding a higher number of segments with a given colouring strategy. We use eight different approaches, the RGB (Red, Green, Blue) colour encodings of which are: white (255, 255, 255); black (0, 0, 0); red (255, 0, 0); green (0, 255, 0); blue (0, 0, 255); pink (255, 192, 203); random – drawn from a uniformly distributed colour space separately for each super-pixel of an individual image; and mean – each segment is occluded with its mean RGB colour. The panels show that the mean occlusion strategy is not as effective at hiding information from the black box as using a single colour for all of the super-pixels (regardless of the colour choice). Similarly, randomising the occlusion colour for each individual segment does not seem to have the detrimental effects observed for the mean colouring. The plots also indicate that when an image is split into more segments, the ineffectiveness of the mean colouring approach gets magnified due to the increased colour uniformity of individual super-pixels. (See Appendix A for the description of our experimental setup.)

B Advantages of Tree-based Interpretable Representations of Tabular Data

Faithfulness of interpretable representations is important for capturing the local behaviour of a black box. In Section 3.2, we argue that this property can be measured by investigating purity of hyper-rectangles that constitute an IR. If the underlying task is crisp classification, we can use the *Gini Impurity* ($\mathcal{L}_{\text{G}}$) defined in Equation 1, where H_i is a set of data points and their labels (x, y) situated within the i^{th} hyper-rectangle and C is the set of all the unique labels c.

$$\begin{aligned} p_{H_i}(c) &= \frac{1}{|H_i|} \sum_{(x,y)\in H_i} \mathbb{1}_{y=c} \\ \mathcal{L}_{\text{G}}(H_i) &= \sum_{c\in C} p_{H_i}(c) \;\; (1 - p_{H_i}(c)) \end{aligned} \tag{1}$$

When the task is regression or probabilistic classification (the formula applies to each individual class separately), on the other hand, we can use the *Mean Squared Error* ($\mathcal{L}_{\text{MSE}}$) – defined in Equation 2 – to quantify numerical uniformity of each hyper-rectangle.

$$\begin{aligned} \bar{y}_{H_i} &= \frac{1}{|H_i|} \sum_{(x,y)\in H_i} y \\ \mathcal{L}_{\text{MSE}}(H_i) &= \frac{1}{|H_i|} \sum_{(x,y)\in H_i} (y - \bar{y}_{H_i})^2 \end{aligned} \tag{2}$$

When combining scores for multiple hyper-rectangles to assess the quality $\mathcal{Q}$ of an interpretable representation, we opt for a weighted average of individual scores $\mathcal{L}$ to account for the (possibly unbalanced) distribution of data points among these segments – see Equation 3.

$$\mathcal{Q} = \frac{1}{\sum_{H_i} |H_i|} \sum_{H_i} |H_i| \;\; \mathcal{L}(H_i) \tag{3}$$

This intuition prompted us to propose using decision trees to partition a feature space according to the separation criterion – given by Equations 1 and 2 respectively for classification and regression trees – applied to the target variable. In case of surrogate explainers, this target becomes a collection of black-box predictions for data points drawn from the explained neighbourhood. While our conjectures seem logically coherent, we support them with experiments on four different data sets, two of which are classification tasks and the other two regression:

- wine recognition[1] (classification),
- breast cancer Wisconsin diagnostic[2] (classification),
- Boston house prices[3] (regression), and
- diabetes[4] (regression).

Using these data, we compare *quartile* and *tree-based* interpretable representations in two variants: *global* – computed collectively for all of the data points with respect to the *discretisation* that underpins the IR, and *local* – calculated individually for each data point given its distinct IR. The results of our experiments are depicted in Figure 8, which shows that tree-based IRs require a fraction of the expressiveness (unique encodings in the binary space) used by the quartile-based IRs to achieve a comparable level of hyper-rectangle purity, especially for the local interpretable representations.

Quartile-based Interpretable Representation This IR is based on quartile discretisation of a given data set [Ribeiro *et al.*, 2016]. The discretisation is *global*, i.e., with respect the the whole data set, however each individual instance receives a distinct IR due to the binarisation that follows. Therefore, the global evaluation is computed on the whole *discretised* data set using the formula given by Equation 3. Local IR faithfulness, on the other hand, is calculated individually for each instance in the data set based on its distinctive binary interpretable representation. This validation is performed for a subset of data that, centred around the explained data point, is within the radius of 60 per cent of the maximum Euclidean distance computed between any two instances in the data set, which simulates locality of the IR.

[1] https://archive.ics.uci.edu/ml/datasets/wine

[2] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

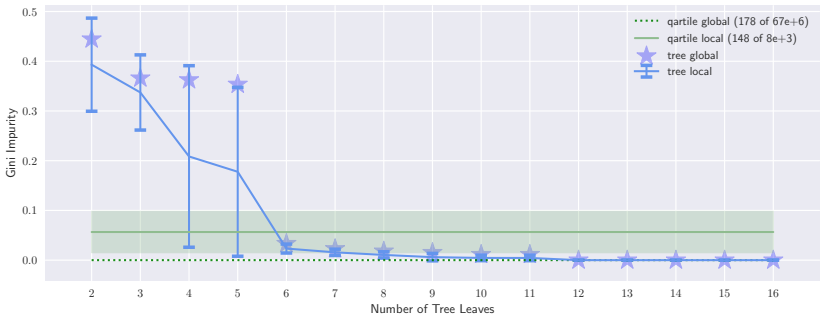
[3] https://archive.ics.uci.edu/ml/machine-learning-databases/housing

[4] https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html

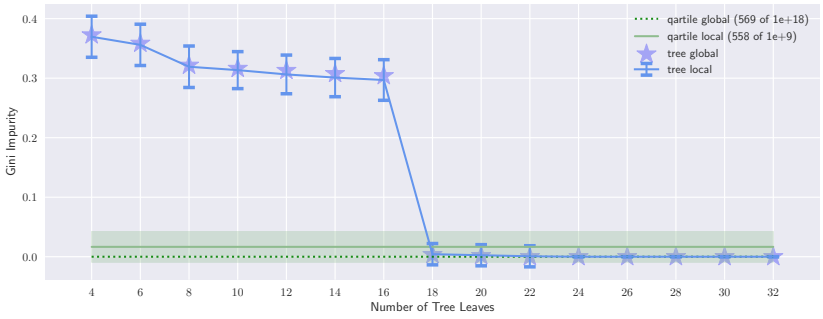
Tree-based Interpretable Representation This IR is based on a partition of the feature space determined by the thresholds learnt with a tree model. The global evaluation is performed by computing purity of the hyper-rectangles created by a tree fitted to the entire data set and validated on this training data, which is a fair comparison given that the quartile-based IR can also access the whole data set. The local IR faithfulness, on the other hand, is calculated independently for each instance in the data set by learning a tree model on a subset of data that, centred around the explained data point, is within the radius of 60 per cent of the maximum Euclidean distance computed between any two instances in the data set, with the same data subset used to compute purity of the resulting hyper-rectangles. Notably, the local method is "disadvantaged" when compared to its quartile-based counterpart since the trees are fitted to a subset of the data, whereas the quartile discretisation has access to all the data.

Interpreting the Results Figure 8 shows the results of our experiments for the aforementioned data sets, each one analysed for a range of different tree widths. The number of tree leaves is directly comparable to the number of unique hyper-rectangles for the global quartile-based discretisation, and their amalgamations for local trees. Each plot depicts the weighted (Equation 3) Gini Impurity or Mean Squared Error, respectively for classification and regression tasks, for all the hyper-rectangles of each individual IR. The dotted green line labelled as "quartile global" is the measure of purity for the quartile discretisation that underlies this type of interpretable representation – it is unique to a data set. The solid green line surrounded by the shading – marked as "quartile local" – corresponds to the mean and standard deviation of hyper-rectangle purity scores of the quartile-based IR for each individual instance in a data set. Equivalent measures are taken for the global and local tree-based IRs for a range of tree widths: "tree global" denoted by a blue $\star$ symbol and "tree local" plotted using a blue line with error bars denoting the standard deviation. In all of the plots, a lower number on the y-axis – weighted "impurity" of an IR – is better.

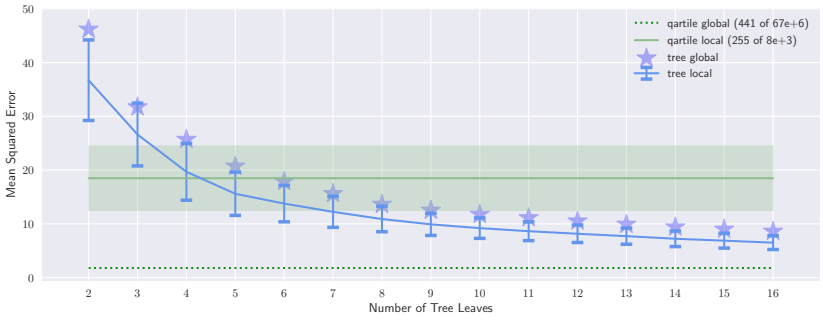
The pair of numbers placed in brackets next to "quartile global" and "quartile local" labels in the legend of each plot indicates how many distinct hyper-rectangles for the global approach, and their combinations for the local approach, are being used by the validation data, out of all the possible unique values that the quartile discretisation and its binarisation can encode. These quantities are directly comparable to the width of trees – recorded on the x-axes of the plots – used to partition the feature space in the tree-based IRs. Given a lack of a black-box model, which predictions should be used to capture the distribution of the target variable for each hyper-rectangle, we utilised the ground truth provided with the aforementioned data sets instead since this substitution does not impede our experiments in any way. All in all, Figure 8 illustrates that interpretable representations created with decision trees are more uniform than their quartile-based competitors, therefore they are superior at capturing the complexity of the underlying labelling mechanism, whatever it may be. Furthermore, they achieve better performance with just a fraction of the encodings used by the other method, i.e., they are more expressive because of the elaborate mechanism used by decision trees to partition and merge the feature space.



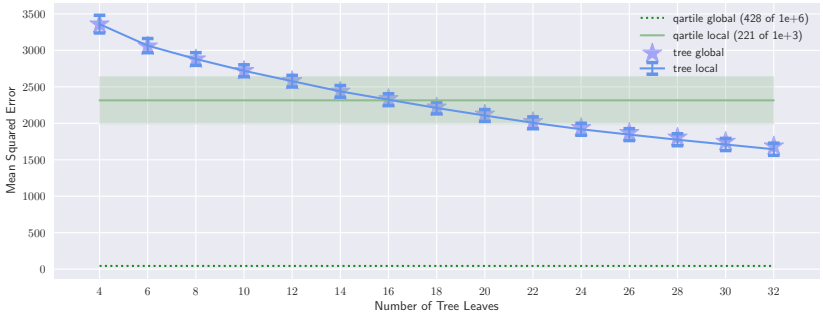
(a) Weighted average of Gini Impurity computed for interpretable representations generated for the *wine* data set.



(b) Weighted average of Gini Impurity computed for interpretable representations generated for the *cancer* data set.



(c) Weighted average of Mean Squared Error computed for interpretable representations generated for the *housing* data set.



(d) Weighted average of Mean Squared Error computed for interpretable representations generated for the *diabetes* data set.

Figure 8: Interpretable representations based on decision trees achieve higher purity of hyper-rectangles (y-axis, lower is better) with a fewer encodings (x-axis), i.e., they are more flexible and expressive. The number of unique encodings used by quartile-based IRs is constant for a data set and is displayed in the legend (presented as the number of encodings used, out of the theoretical limit supported by the representation); whereas for tree-based IRs, it is equivalent to the number of leaves, which is recorded on the x-axis. Panels (c) and (d) do not capture the tree width at which this IR *globally* outperforms the quartile-based IR, which is 80 (compared to 441) and 224 (compared to 428) respectively for the *housing* and *diabetes* data sets. For more details, see the "Interpreting the Results" paragraph in Appendix B.

C Analytical Solution to OLS for Binary Interpretable Representations of Tabular Data

Linear models can be used to quantify importance of interpretable concepts for black-box predictions [Friedman *et al.*, 2008; Ribeiro *et al.*, 2016]. However, for some binary interpretable domains, such an approach is inherently flawed. In this appendix, we show how the interpretable concept importance measured by coefficients of a linear model may be deceiving. This phenomenon is particularly prominent for tabular data transformed into the binary interpretable representation introduced in Section 3. The insights stemming from our analysis can be used to manipulate surrogate explanations, e.g., those produced with LIME [Ribeiro *et al.*, 2016], by using specially crafted, yet perfectly valid, IR discretisation and data sample.

Our results are based on the analytical solution to unweighted (Θ) and weighted ($\Theta_{\mathbf{W}}$) OLS outlined in the equations below, where $\mathbf{X}$ is the binary interpretable representation data matrix, $\mathbf{W}$ is the weight matrix, and $\mathbf{y}$ is the black-box predictions vector. In our analysis, we assume that the black box is a probabilistic classifier, in which case $\mathbf{y}$ holds probabilities of the explained class, however a similar line of reasoning applies to regressors and crisp classifiers. In the latter case, the predictions are assumed to be 1 if they are the same as the class assigned to the explained instance by the black box, and 0 for any other class. Modelling $\mathbf{y}$ in such a way is equivalent to generating one-vs-rest explanations of probabilistic black boxes since they only model probabilities of the explained class. Nonetheless, both approaches measure the importance of interpretable concepts – using the coefficients of linear models – for a selected class when tasked with telling it apart from the other classes.

$$\Theta = (\mathbf{X}^T\mathbf{X})^{-1}\,\mathbf{X}^T\mathbf{y}$$
$$\Theta_{\mathbf{W}} = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\,\mathbf{X}^T\mathbf{W}\mathbf{y}$$

In the interest of brevity and readability, we analyse tabular data with two numerical features – akin to the example shown in Figure 6 – but our findings generalise to an arbitrary number of attributes that are both categorical and numerical. In a generic setting, for n features there will be n binary concepts with 2^n unique encodings (cardinality) in the interpretable representation. If additionally we choose to model the intercept of the linear model, a phantom all-1 column vector is inserted at the front of the data matrix $\mathbf{X}$. Therefore, the $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}^T\mathbf{W}\mathbf{X}$ components of Θ and $\Theta_{\mathbf{W}}$ respectively are square matrices of $n \times n$ shape sans the intercept or $(n+1) \times (n+1)$ when the intercept is modelled.

Figure 6 depicts a simplistic view of sampling for two numerical features with just one data point in each hyper-rectangle, but in reality, we should expect their large number since it allows to better approximate the behaviour of the black box, especially when the number of features is high. In this particular case, the binary interpretable representation data matrix $\mathbf{X}$ – with the first column (red) inserted for modelling the intercept and the remaining columns (blue) representing the binary data – is:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

which gives:

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 9 & 3 & 3 \\ 3 & 3 & 1 \\ 3 & 1 & 3 \end{bmatrix}.$$

Since some of the hyper-rectangles are merged when transitioning from the discrete into the binary interpretable representation, $\mathbf{X}$ contains duplicated rows. The influence of this phenomenon is magnified even more when multiple data points are sampled within a single hyper-rectangle. Without loss of generality, we can use the *weighted* variant of OLS, with the data set $\mathbf{X}$ containing only one copy of each unique binary data point and the weights corresponding to their counts. In this case:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} w_{11} & 0 & 0 & 0 \\ 0 & w_{10} & 0 & 0 \\ 0 & 0 & w_{01} & 0 \\ 0 & 0 & 0 & w_{00} \end{bmatrix},$$

where w_{ij} is the count of data points residing in all of the hyper-rectangles that are assigned the (i, j) coordinates in the binary interpretable representation – see the $(x^\star, y^\star)$ coordinates in Figure 6 for reference. Therefore, for an arbitrary number of data points with two numerical features when modelling the intercept:

$$\begin{aligned}
\mathbf{X}^T\mathbf{W}\mathbf{X} &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} w_{11} & 0 & 0 & 0 \\ 0 & w_{10} & 0 & 0 \\ 0 & 0 & w_{01} & 0 \\ 0 & 0 & 0 & w_{00} \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} w_{11} & w_{10} & w_{01} & w_{00} \\ w_{11} & w_{10} & 0 & 0 \\ w_{11} & 0 & w_{01} & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} \sum w_{ij} & w_{11} + w_{10} & w_{11} + w_{01} \\ w_{11} + w_{10} & w_{11} + w_{10} & w_{11} \\ w_{11} + w_{01} & w_{11} & w_{11} + w_{01} \end{bmatrix}.
\end{aligned}$$

For the example in Figure 6 – where $w_{11} = 1$, $w_{10} = 2$, $w_{01} = 2$ and $w_{00} = 4$ – a calculation for the weighted variant agrees with the previous result computed directly for $\mathbf{X}^T\mathbf{X}$.

Next, we analyse the second component of the $\Theta_\mathbf{W}$ formula:

$$\begin{aligned}
\mathbf{X}^T\mathbf{W}\mathbf{y} &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} w_{11} & 0 & 0 & 0 \\ 0 & w_{10} & 0 & 0 \\ 0 & 0 & w_{01} & 0 \\ 0 & 0 & 0 & w_{00} \end{bmatrix} \times \begin{bmatrix} y_{11} \\ y_{10} \\ y_{01} \\ y_{00} \end{bmatrix} \\
&= \begin{bmatrix} w_{11} & w_{10} & w_{01} & w_{00} \\ w_{11} & w_{10} & 0 & 0 \\ w_{11} & 0 & w_{01} & 0 \end{bmatrix} \times \begin{bmatrix} y_{11} \\ y_{10} \\ y_{01} \\ y_{00} \end{bmatrix} \\
&= \begin{bmatrix} \sum w_{ij}y_{ij} \\ w_{11}y_{11} + w_{10}y_{10} \\ w_{11}y_{11} + w_{01}y_{01} \end{bmatrix}.
\end{aligned}$$

However, this formulation assumes that all of the data points that share the same coordinates in the binary interpretable representation have the same label y. To allow multiple copies of the same data point with distinct labels, we generalise this result by going back to Θ, which is the solution to the classic OLS. This approach is valid since the weighted OLS for which the weights represent the count of each unique data point is equivalent to the classic OLS for a data set whose instances are duplicated according to the counts given by the corresponding weights.

Let us denote $f : \mathcal{X} \to \mathcal{Y}$ as the black-box model and $IR : \mathcal{X} \to \mathcal{X}^\star$ as the transformation function from tabular data $\mathcal{X}$ into their binary interpretable representation $\mathcal{X}^\star$. Let us further define $\mathcal{W}_{ij} = \{x \in \mathbf{X} : IR(x) = (i, j)\}$ as the set of all the data points that are assigned the same interpretable binary representation (i, j), and $\mathcal{W} = \mathbf{X}$ as the set of all the data points. Now, recall that w_{ij} is the count of data points whose binary interpretable representation is (i, j), therefore $|\mathcal{W}_{ij}| = w_{ij}$ and $|\mathcal{W}| = \sum w_{ij}$. Without loss of generality, we can reformulate the $\mathbf{X}^T\mathbf{W}\mathbf{y}$ part of the $\Theta_\mathbf{W}$ equation as $\mathbf{X}^T\mathbf{y}$ in order to sum over all of the black-box predictions for their respective hyper-rectangles:

$$\mathbf{X}^T\mathbf{y} = \begin{bmatrix} \sum w_{ij}y_{ij} \\ w_{11}y_{11} + w_{10}y_{10} \\ w_{11}y_{11} + w_{01}y_{01} \end{bmatrix} = \begin{bmatrix} \sum_{i \in \mathcal{W}} y_i \\ \sum_{i \in \mathcal{W}_{11} \cup \mathcal{W}_{10}} y_i \\ \sum_{i \in \mathcal{W}_{11} \cup \mathcal{W}_{01}} y_i \end{bmatrix}.$$

This step allows us to relax the assumption of duplicated labels without loss of generality, hence not impose restrictions on whether the black box is probabilistic and whether the discrete representation has perfect fidelity with respect to the black box.

Finally, to better understand the meaning of importance-based explanations, we reformulate the sum of black-box predictions

into their average:

$$\mathbf{X}^T\mathbf{y} = \begin{bmatrix} \sum_{i\in\mathcal{W}} y_i \\ \sum_{i\in\mathcal{W}_{11}\cup\mathcal{W}_{10}} y_i \\ \sum_{i\in\mathcal{W}_{11}\cup\mathcal{W}_{01}} y_i \end{bmatrix} = \begin{bmatrix} \sum_{i\in\mathcal{W}} y_i / \sum w_{ij} * \sum w_{ij} \\ \sum_{i\in\mathcal{W}_{11}\cup\mathcal{W}_{10}} y_i/(w_{11}+w_{10}) * (w_{11}+w_{10}) \\ \sum_{i\in\mathcal{W}_{11}\cup\mathcal{W}_{01}} y_i/(w_{11}+w_{01}) * (w_{11}+w_{01}) \end{bmatrix} = \begin{bmatrix} \bar{y}_{\mathcal{W}} * \sum w_{ij} \\ \bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{10}} * (w_{11}+w_{10}) \\ \bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{01}} * (w_{11}+w_{01}) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \bar{y}_{\mathcal{W}} * \sum w_{ij} \\ \bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{10}} * (w_{11}+w_{10}) \\ \bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{01}} * (w_{11}+w_{01}) \end{bmatrix}$$

$$= \begin{bmatrix} \sum w_{ij} & 0 & 0 \\ 0 & w_{11}+w_{10} & 0 \\ 0 & 0 & w_{11}+w_{01} \end{bmatrix} \times \begin{bmatrix} \bar{y}_{\mathcal{W}} \\ \bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{10}} \\ \bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{01}} \end{bmatrix},$$

and combine this result with $\mathbf{X}^T\mathbf{W}\mathbf{X}$:

$$\begin{bmatrix} \sum w_{ij} & w_{11}+w_{10} & w_{11}+w_{01} \\ w_{11}+w_{10} & w_{11}+w_{10} & w_{11} \\ w_{11}+w_{01} & w_{11} & w_{11}+w_{01} \end{bmatrix}^{-1} \times \begin{bmatrix} \sum w_{ij} & 0 & 0 \\ 0 & w_{11}+w_{10} & 0 \\ 0 & 0 & w_{11}+w_{01} \end{bmatrix} \times \begin{bmatrix} \bar{y}_{\mathcal{W}} \\ \bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{10}} \\ \bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{01}} \end{bmatrix}$$

$$= \begin{bmatrix} \sum w_{ij} & w_{11}+w_{10} & w_{11}+w_{01} \\ w_{11}+w_{10} & w_{11}+w_{10} & w_{11} \\ w_{11}+w_{01} & w_{11} & w_{11}+w_{01} \end{bmatrix}^{-1} \times \begin{bmatrix} \frac{1}{\sum w_{ij}} & 0 & 0 \\ 0 & \frac{1}{w_{11}+w_{10}} & 0 \\ 0 & 0 & \frac{1}{w_{11}+w_{01}} \end{bmatrix}^{-1} \times \begin{bmatrix} \bar{y}_{\mathcal{W}} \\ \bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{10}} \\ \bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{01}} \end{bmatrix}$$

$$= \left(\begin{bmatrix} \frac{1}{\sum w_{ij}} & 0 & 0 \\ 0 & \frac{1}{w_{11}+w_{10}} & 0 \\ 0 & 0 & \frac{1}{w_{11}+w_{01}} \end{bmatrix} \times \begin{bmatrix} \sum w_{ij} & w_{11}+w_{10} & w_{11}+w_{01} \\ w_{11}+w_{10} & w_{11}+w_{10} & w_{11} \\ w_{11}+w_{01} & w_{11} & w_{11}+w_{01} \end{bmatrix} \right)^{-1} \times \begin{bmatrix} \bar{y}_{\mathcal{W}} \\ \bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{10}} \\ \bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{01}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \frac{w_{11}+w_{10}}{\sum w_{ij}} & \frac{w_{11}+w_{01}}{\sum w_{ij}} \\ 1 & 1 & \frac{w_{11}}{w_{11}+w_{10}} \\ 1 & \frac{w_{11}}{w_{11}+w_{10}} & 1 \end{bmatrix}^{-1} \times \begin{bmatrix} \bar{y}_{\mathcal{W}} \\ \bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{10}} \\ \bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{01}} \end{bmatrix}.$$

This outcome allows us to draw conclusions about the meaning of interpretable concept importance given by the coefficients of a surrogate linear model when the intercept is modelled (red & blue) and without it (blue). In particular, the importance of interpretable concepts is *solely* based on:

- **the proportion determined by the number of the data points** residing in the explained hyper-rectangle ($\mathcal{W}_{11}$) divided by the hyper-rectangles aligned with the explained hyper-rectangle along every axis: $\mathcal{W}_{11} \cup \mathcal{W}_{10}$ for the first feature and $\mathcal{W}_{11} \cup \mathcal{W}_{01}$ for the second; and
- **the average value predicted** in the latter two subspaces – $\mathcal{W}_{11} \cup \mathcal{W}_{10}$ and $\mathcal{W}_{11} \cup \mathcal{W}_{01}$ – by the black box (scaled appropriately when the intercept is modelled).

For example, consider Figure 9 where $x_1^\star$ denotes the first binary interpretable feature and $x_2^\star$ the second. In this case, $\mathcal{W}_{11}$ is the yellow hyper-rectangle; $\mathcal{W}_{11} \cup \mathcal{W}_{10}$ is the union of the yellow and green hyper-rectangles; and $\mathcal{W}_{11} \cup \mathcal{W}_{01}$ is the combination of yellow and blue hyper-rectangles. Finally, $\bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{10}}$ is the average prediction in the vertical green&yellow segment, and $\bar{y}_{\mathcal{W}_{11}\cup\mathcal{W}_{01}}$ is the average prediction in the horizontal blue&yellow stripe.

The intercept value is additionally determined by:

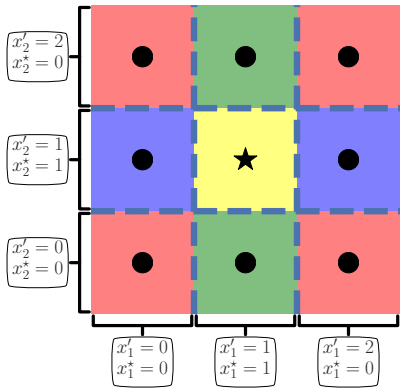


Figure 9: Example of discrete (x_1', x_2') and binary interpretable $(x_1^\star, x_2^\star)$ representations of tabular data. $\star$ represents the explained instance.

- *the proportion* given by the number of data points in the hyper-rectangles aligned with the explained hyper-rectangle along every axis divided by the total number of data points; and
- *the average* value predicted by the black box for all the data points.

Intuitively, the instances not aligned with the explained hyper-rectangle – red blocks in Figure 9 – are assigned the $(0, 0)$ coordinates in the binary interpretable representation, therefore they cannot contribute to the feature coefficients of a linear model, just the intercept. This can be easily seen with $g(\mathbf{x}; \Theta) = \sum_{i=0}^{n} \Theta_i x_i$ formula, where $x_0 = 1$ is the phantom feature and the remaining data features $x_1, \ldots, x_n$ are 0.

An important insight uncovered by these results is **partial irrelevance of the discretisation quality** given a fixed number of data points placed in the identified collections of hyper-rectangles. Using this property, we can manipulate the explanation by altering the number of data points in relevant partitions, with the discretisation faithfulness having relatively small influence. For example, consider the two discretisations depicted earlier in Figure 5, assuming that the explained hyper-rectangle is $(x', y') = (1, 1)$ for both sub-plots and that the $(x', y') = (1, 0)$ and $(x', y') = (1, 1)$ partitions in Figure 5b have *three* additional data points each. In this case, when modelling the importance of interpretable components without the intercept, the only difference between these two sub-plots are the predictions of the instances placed in the expanded hyper-rectangles, i.e., $(x', y') = (1, 0)$ and $(x', y') = (1, 1)$, since:

Figure 5a $w_{11} = 4$, $w_{01} = 4 + 4 = 8$ and $w_{10} = 4$, leading to $\frac{w_{11}}{w_{11}+w_{10}} = \frac{4}{4+4} = \frac{1}{2}$ and $\frac{w_{11}}{w_{11}+w_{01}} = \frac{4}{4+8} = \frac{1}{3}$; and

Figure 5b $w_{11} = 2 + \mathit{3} = 5$, $w_{01} = 4 + 4 + 2 = 10$ and $w_{10} = 2 + \mathit{3} = 5$, leading to $\frac{w_{11}}{w_{11}+w_{10}} = \frac{5}{5+5} = \frac{1}{2}$ and $\frac{w_{11}}{w_{11}+w_{01}} = \frac{5}{5+10} = \frac{1}{3}$.

Depending on the gradient smoothness of the probabilistic black box, these explanations may slightly differ. However, if the additional six data points are placed such that the average black-box predictions of $\mathcal{W}_{11} \cup \mathcal{W}_{10}$ and $\mathcal{W}_{11} \cup \mathcal{W}_{01}$ are identical for both discretisations, the resulting explanations will be the same. Alternatively, if they are class predictions instead of probabilities, the two explanations will also be indistinguishable regardless of where the additional six instances are situated within their respective hyper-rectangles. The added benefit of this observation is evidence that partitioning each numerical feature into more than three splits is not beneficial, with the most important bin boundaries being the ones enclosing the explained data point.